



Proyecto Final

Inventario en Línea

Camargo Badillo Luis Mauricio
Gudiño Romero Miguel Angel

20 de mayo de 2025

Temas Selectos de Computación II
Desarrollo de APIs

David Bryan Padilla Alemán

Matemáticas Aplicadas y Computación

Índice

Objetivo	2
Problema por Resolver	2
Propuesta de Solución	2
Diseño de la Solución	3
Casos de Uso	3
Componentes	3
Interacción entre Componentes	3
Modelo de Datos	4
Tecnologías Utilizadas	4
Ejecución del Proyecto con Docker	5
Requisitos Previos	5
Instrucciones	5
Casos de Ejecución	5
Autenticación	6
Inventario	8
Conclusión	19

En la actualidad, la gestión eficiente de inventarios es un reto común para empresas y organizaciones de todos los tamaños. Muchos negocios aún dependen de métodos manuales o sistemas obsoletos, lo que genera errores, falta de visibilidad y dificultades para mantener información actualizada y confiable. La mejora de estos procesos es fundamental para mejorar la organización interna y reducir errores humanos.

Este proyecto propone una solución basada en una aplicación web full-stack que permite a los usuarios gestionar sus inventarios de manera eficiente y segura. La aplicación está diseñada para facilitar el registro, consulta, actualización y eliminación de productos en tiempo real, ofreciendo una herramienta moderna y accesible para la administración de inventarios.

Objetivo

El objetivo principal de este proyecto es diseñar y desarrollar una aplicación web que permita la gestión eficiente de inventarios en línea. Esta solución busca proporcionar a los usuarios una herramienta moderna, accesible y segura para administrar los productos de una tienda o empresa, facilitando el registro, consulta, actualización y eliminación de los artículos en tiempo real. El sistema estará orientado a mejorar la organización interna, reducir errores humanos y optimizar los procesos relacionados con el manejo de inventarios.

Problema por Resolver

Actualmente, muchas organizaciones enfrentan retos significativos en la administración de sus inventarios, especialmente cuando dependen de métodos manuales o sistemas poco actualizados. Entre los principales problemas se encuentran la falta de visibilidad sobre el estado actual del inventario, la dificultad para registrar y consultar movimientos de productos (altas, bajas, modificaciones y búsquedas), la posibilidad de cometer errores en el conteo o registro, y la ausencia de reportes claros y exportables que respalden las operaciones realizadas.

Propuesta de Solución

Para abordar el problema identificado, se desarrolló una aplicación web full-stack que integra tecnologías modernas tanto en el frontend como en el backend. El frontend fue implementado con React, permitiendo una interfaz de usuario dinámica, intuitiva y visualmente atractiva. La página principal presenta de manera clara las funcionalidades disponibles y cuenta con formularios de inicio de sesión y registro para garantizar la seguridad.

El backend se implementó utilizando Java Spring Boot, creando una API REST que permite la comunicación entre el frontend y la base de datos MySQL. Esta arquitectura facilita la escalabilidad y el mantenimiento del sistema, además de asegurar la integridad y persistencia de los datos.

Una vez autenticados, los usuarios pueden realizar todas las operaciones fundamentales del inventario: agregar nuevos productos, consultar el listado y detalles de los artículos, eliminar productos existentes o modificar la cantidad y características de los mismos. Cada operación se refleja en tiempo real en la interfaz y el sistema proporciona la opción de descargar un reporte en PDF con los resultados de la operación realizada. Así, se garantiza que los usuarios

cuenten con información actualizada y respaldada, mejorando la eficiencia y confiabilidad en la gestión de inventarios.

Diseño de la Solución

Este sistema de gestión de inventarios fue diseñado bajo una arquitectura web full-stack, separando claramente la interfaz de usuario (frontend), la API (backend), y la persistencia de datos (base de datos).

Casos de Uso

El sistema tiene los siguientes casos de uso principales:

- Registro y autenticación de usuarios.
- Alta, baja, modificación y consulta de productos en el inventario.
- Exportación de reportes en PDF.

El sistema implementa autenticación basada en JWT (JSON Web Tokens) para proteger los endpoints sensibles y garantizar que solo los usuarios autorizados puedan realizar operaciones sobre el inventario.

Componentes

El sistema se compone de tres componentes principales:

- **Frontend:** Aplicación desarrollada en React, responsable de la interacción con el usuario, validación de formularios y consumo de la API REST.
- **Backend:** API REST construida con Java Spring Boot, encargada de la lógica, autenticación, autorización y comunicación con la base de datos.
- **Base de Datos:** Servidor MySQL que almacena la información de usuarios, productos, inventario y operaciones.

Interacción entre Componentes

- El usuario accede a la aplicación web desde su navegador y realiza acciones como iniciar sesión, registrar productos o consultar el inventario.
- El frontend envía solicitudes HTTP a la API REST del backend, que valida y procesa las peticiones.
- El backend interactúa con la base de datos MySQL para almacenar o recuperar información según la operación solicitada.
- Las respuestas del backend son procesadas por el frontend, que actualiza la interfaz de usuario en tiempo real.

- Para ciertas operaciones, el usuario puede exportar la información visualizada en la interfaz a un archivo PDF.

A continuación, los diversos diagramas que ilustran la interacción entre estos componentes para los distintos casos de uso:

Modelo de Datos

La base de datos relacional está compuesta por las siguientes tablas principales:

- **users:** Almacena la información de los usuarios registrados, incluyendo credenciales.
- **items:** Contiene los datos de los productos disponibles, incluyendo nombre, descripción, precio y cantidad.
- **operations:** Guarda un historial de las operaciones realizadas sobre el inventario.

Tecnologías Utilizadas

El desarrollo del sistema requirió la integración de diversas tecnologías modernas que se seleccionaron en función a su facilidad de uso y compatibilidad con los objetivos del proyecto.

- **React:** Biblioteca de JavaScript utilizada para construir la interfaz de usuario (frontend). Permite crear componentes reutilizables y una experiencia dinámica e interactiva para el usuario.
- **Vite:** Herramienta de construcción y desarrollo rápido para proyectos frontend en React, utilizada para optimizar el flujo de trabajo y la velocidad de desarrollo.
- **Mantine UI:** Biblioteca de componentes de interfaz para React, empleada para lograr una apariencia moderna en la aplicación, sin tener que desarrollar los componentes desde cero.
- **Java Spring Boot:** Framework para el desarrollo del backend, encargado de la exposición de la API REST y la gestión de la seguridad y autenticación.
- **MySQL:** Sistema de gestión de bases de datos relacional, utilizado para almacenar de manera persistente la información de usuarios, productos y operaciones.
- **JWT (JSON Web Tokens):** Tecnología de autenticación y autorización, utilizada para proteger los endpoints sensibles.
- **Docker:** Plataforma de contenedores empleada para facilitar el despliegue y la ejecución del sistema en cualquier entorno, asegurando la portabilidad.
- **Docker Compose:** Herramienta para orquestar múltiples contenedores de Docker (frontend, backend y base de datos) y simplificar su despliegue.
- **jsPDF:** Librería de JavaScript utilizada en el frontend para exportar los reportes de inventario en formato PDF.

- **Axios:** Cliente HTTP para JavaScript, empleado en el frontend para realizar solicitudes a la API REST del backend.

Ejecución del Proyecto con Docker

Para facilitar la ejecución y despliegue del sistema, se utilizó Docker. Esto permite levantar todos los servicios necesarios (frontend, backend y base de datos) con un solo comando, sin preocuparse por dependencias o configuraciones específicas del entorno local.

Requisitos Previos

- Tener instalado Docker y Docker Compose en el sistema.
- No tener otros servicios ocupando los puertos 80, 8080 o 3306.

Instrucciones

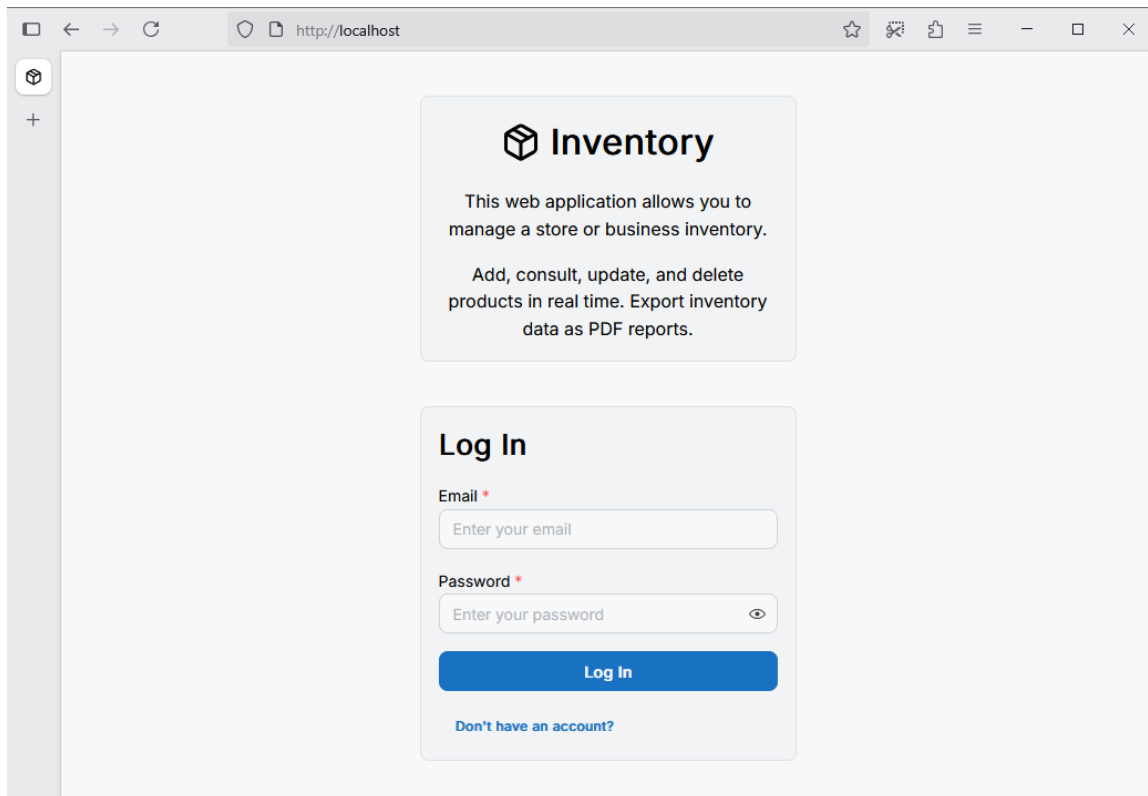
Desde la raíz del proyecto, ejecutar:

```
docker-compose up --build
```

Esto construirá y levantará todos los servicios. La aplicación frontend estará disponible en <http://localhost>, el backend en <http://localhost:8080> y el servidor MySQL en el puerto 3306 (localhost:3306).

Casos de Ejecución

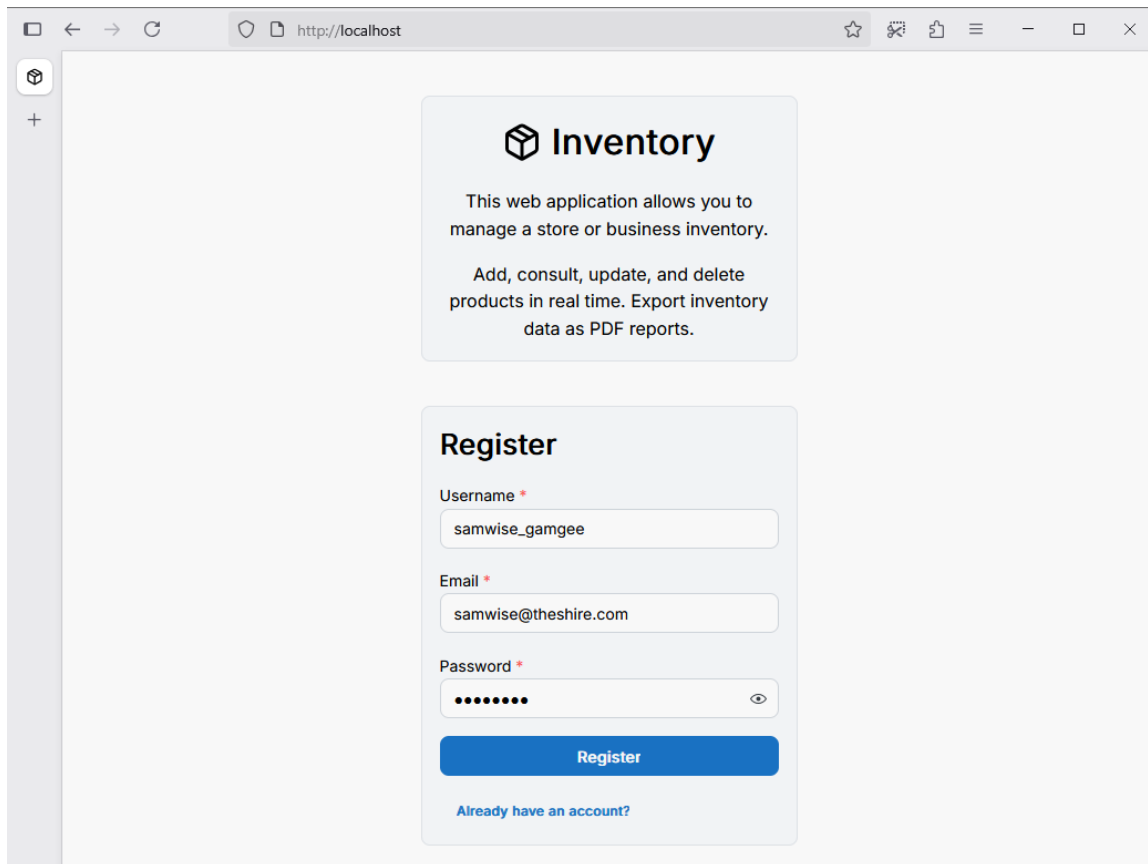
Al ingresar a la URL de la aplicación web, se llega a la pantalla inicial, donde se muestra el título del proyecto, una breve descripción de su funcionalidad y, más abajo, un cuadro de autenticación donde el usuario puede iniciar sesión, si ya tiene una cuenta, o registrarse si no la tiene todavía.



Autenticación

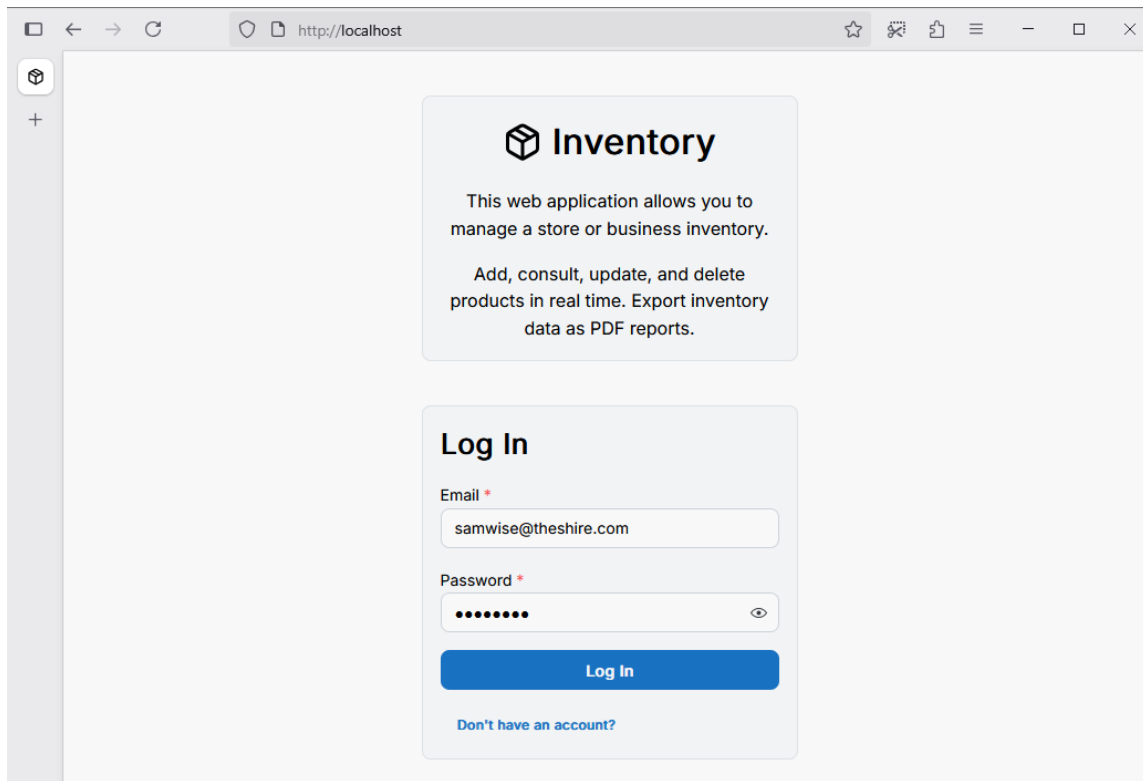
Registro

Inicialmente, no existe ningún usuario registrado, así que se debe registrar uno nuevo. Para ello, se da clic en el botón “Don't have an account?” y luego se debe completar el formulario de registro con un nombre de usuario, un correo electrónico y una contraseña. El usuario quedará registrado una vez que se haga clic en el botón “Register”, redirigiendo a la pantalla de inventario que se verá más adelante.



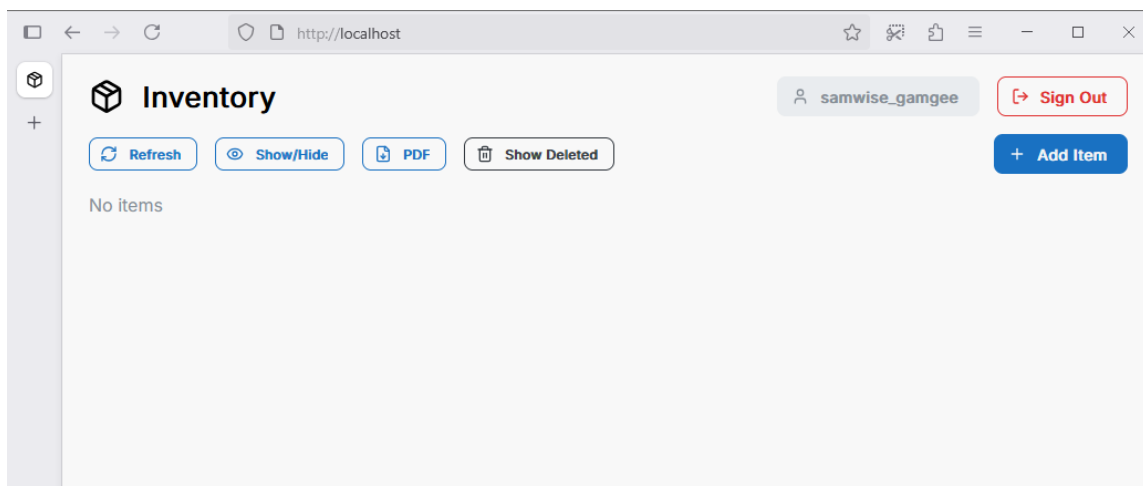
Inicio de Sesión

Cuando un usuario ya está registrado, puede iniciar sesión simplemente ingresando su correo electrónico y contraseña en el formulario correspondiente, que aparece por defecto al ingresar la aplicación. Al hacer clic en el botón “Login”, se redirigirá a la pantalla de inventario. Para acceder a este formulario desde el formulario de registro, basta con darle clic al botón “Already have an account?”.



Inventario

Después de registrarse o iniciar sesión, se redirige al usuario a la pantalla de inventario.



En esta pantalla es donde se lleva a cabo la gran parte de la funcionalidad del sistema. En la parte superior, se encuentra el nombre de la aplicación, el nombre del usuario actualmente autenticado y un botón para cerrar sesión. En la parte inferior, se muestra una tabla con los ítems que se encuentran en el inventario, que inicialmente está vacía.

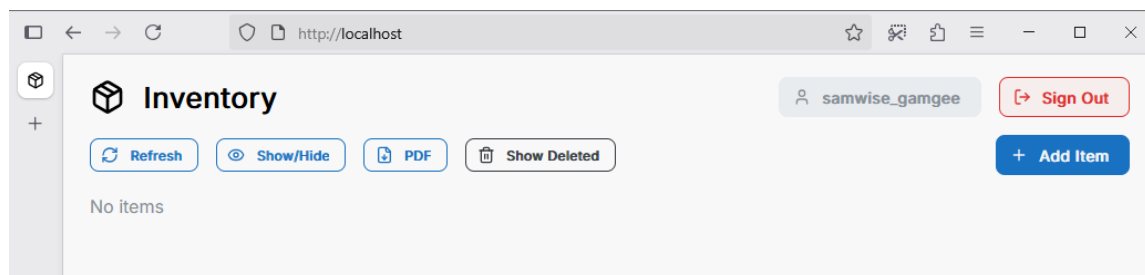
Arriba de la tabla, aparecen algunos controles para manipularla o interactuar con el sistema. Estos controles son, de izquierda a derecha:

- Un botón para actualizar la vista de la tabla.

- Un botón para elegir qué campos se desean mostrar en la tabla.
- Un botón para exportar la vista actual de la tabla a un archivo PDF.
- Un botón para mostrar los ítems eliminados.
- Un botón para agregar un nuevo ítem al inventario.

Cierre de Sesión

Para cerrar sesión, basta con darle clic al botón “Sign Out”, lo que redirigirá a la pantalla inicial donde los usuarios se pueden autenticar.



Agregar ítem

Para agregar un nuevo ítem al inventario, se debe hacer clic en el botón “Add Item”. Esto abrirá un formulario donde se deben completar los campos requeridos: nombre, SKU, descripción, precio y cantidad. Una vez completados, se debe hacer clic en el botón “Add” para guardar el nuevo ítem en la base de datos.

The screenshot shows a web browser at `http://localhost` displaying an 'Inventory' application. A modal titled 'Add new item' is open, with the 'Interactive' tab selected. The form contains the following fields:

- Name ***: The One Ring
- SKU ***: LOTR-001
- Description**: One ring to rule them all.
- Price ***: 9999.99
- Quantity ***: 1

At the bottom of the modal are 'Cancel' and 'Add' buttons. The background shows a sidebar with a cube icon and a main area with buttons for 'Refresh', 'Show/Hide', 'PDF', 'Show Deleted', and 'Add Item'. The user 'samwise_gamgee' is logged in.

También existe la opción de insertar directamente estos campos en un formato JSON, seleccionando la pestaña correspondiente.

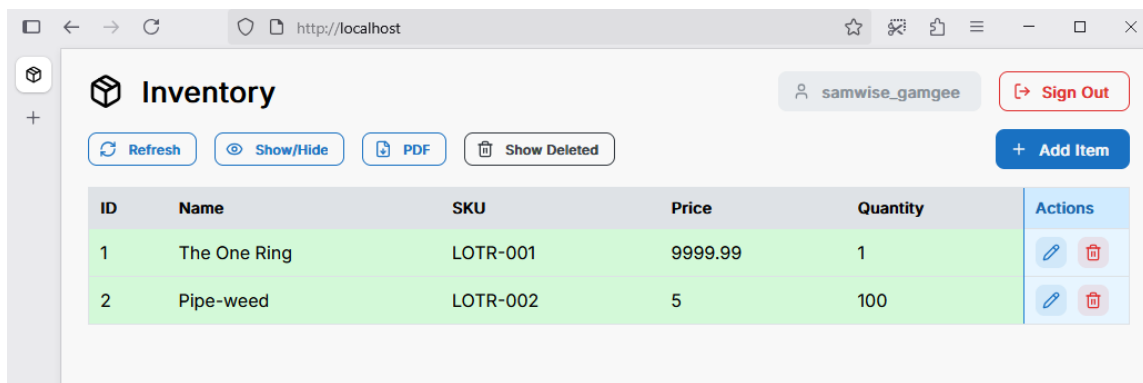
The screenshot shows the same 'Inventory' application, but the 'JSON' tab is selected in the 'Add new item' modal. The 'Item JSON' field contains the following JSON object:





```
{
  "name": "Pipe-weed",
  "sku": "LOTR-002",
  "description": "Popular Hobbit smoking herb.",
  "price": 5.00,
  "quantity": 100
}
```

The 'Cancel' and 'Add' buttons are at the bottom. In the background, a table is visible with the following data:

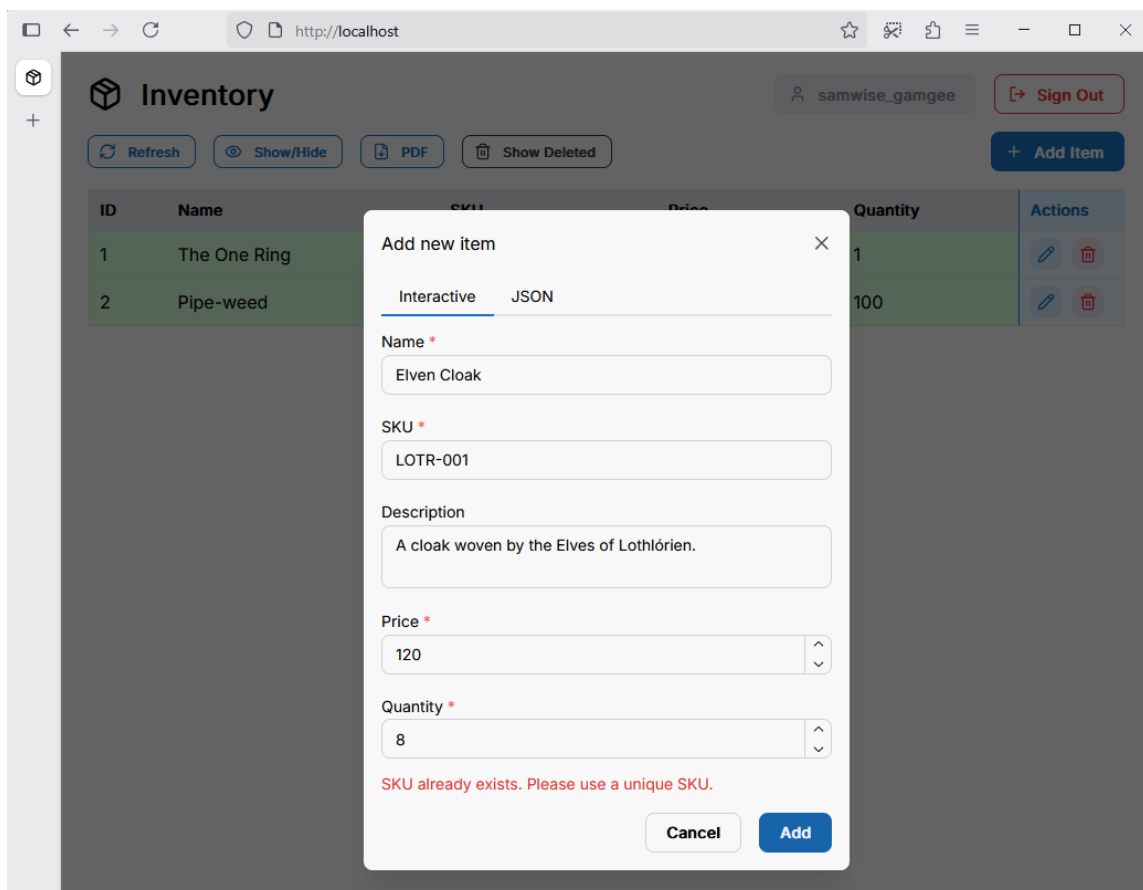
ID	Name	SKU	Price	Quantity	Actions
1	The One Ring	LOTR-001	9999.99	1	[Edit] [Delete]

Los ítems agregados aparecen en color verde.



ID	Name	SKU	Price	Quantity	Actions
1	The One Ring	LOTR-001	9999.99	1	 
2	Pipe-weed	LOTR-002	5	100	 

Si existe algún impedimento para agregar el ítem, se le notificará al usuario. Por ejemplo, si el SKU ya existe o si el JSON no es válido.



Add new item

Interactive JSON

Name *

Elven Cloak

SKU *

LOTR-001

Description

A cloak woven by the Elves of Lothlórien.

Price *

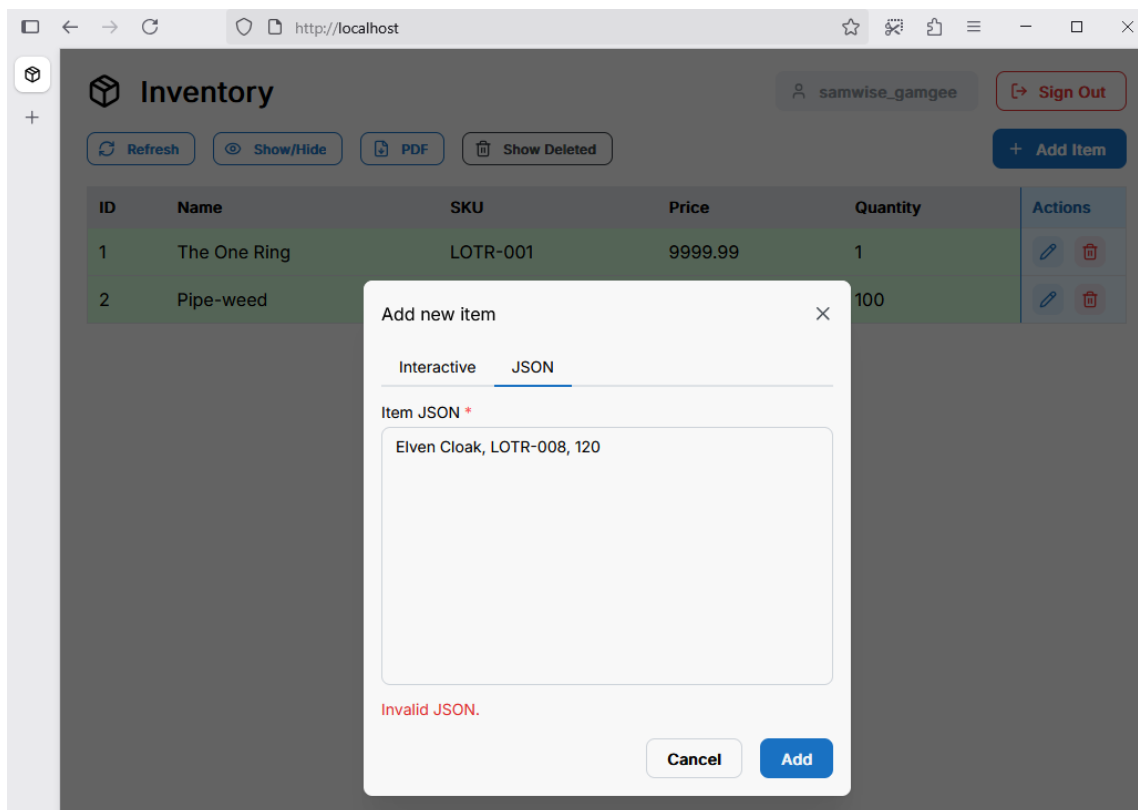
120

Quantity *

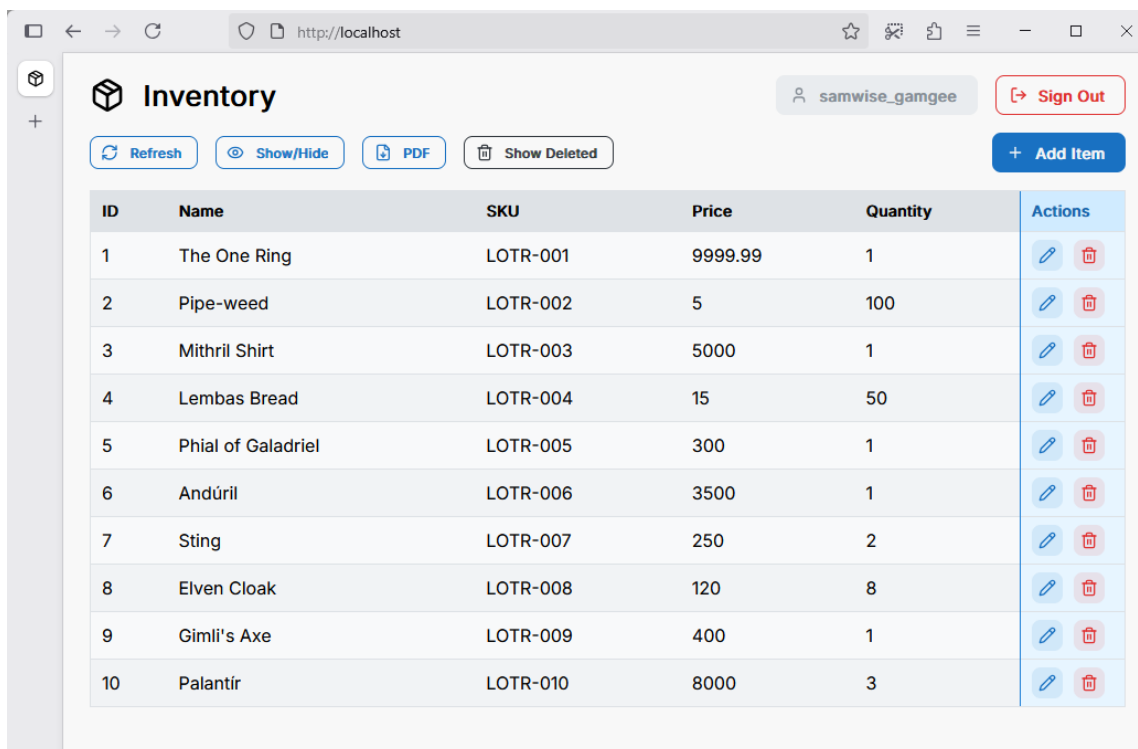
8

SKU already exists. Please use a unique SKU.

Cancel Add

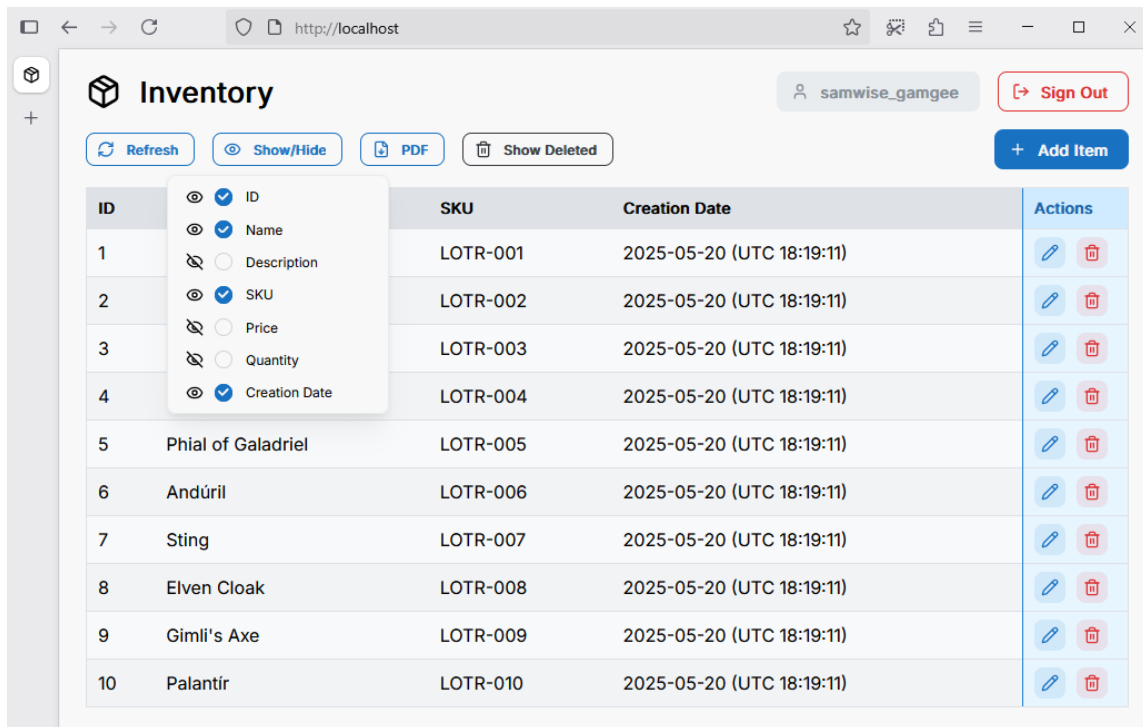


Una vez que ya se tiene al menos un ítem en el inventario, al final de cada fila aparecen dos botones: uno para editar y otro para eliminar.



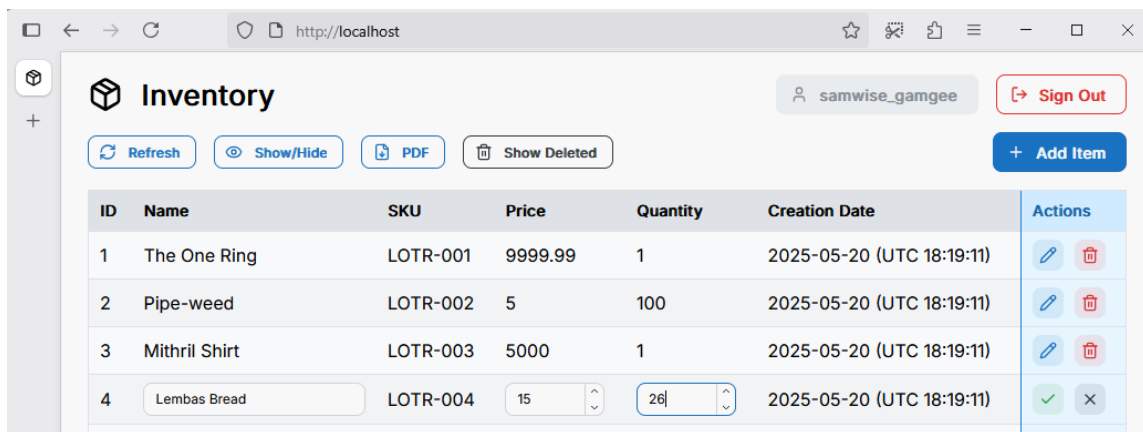
Elegir Columnas Mostradas

Es posible darle clic al botón "Show/Hide" para cambiar las columnas mostradas.

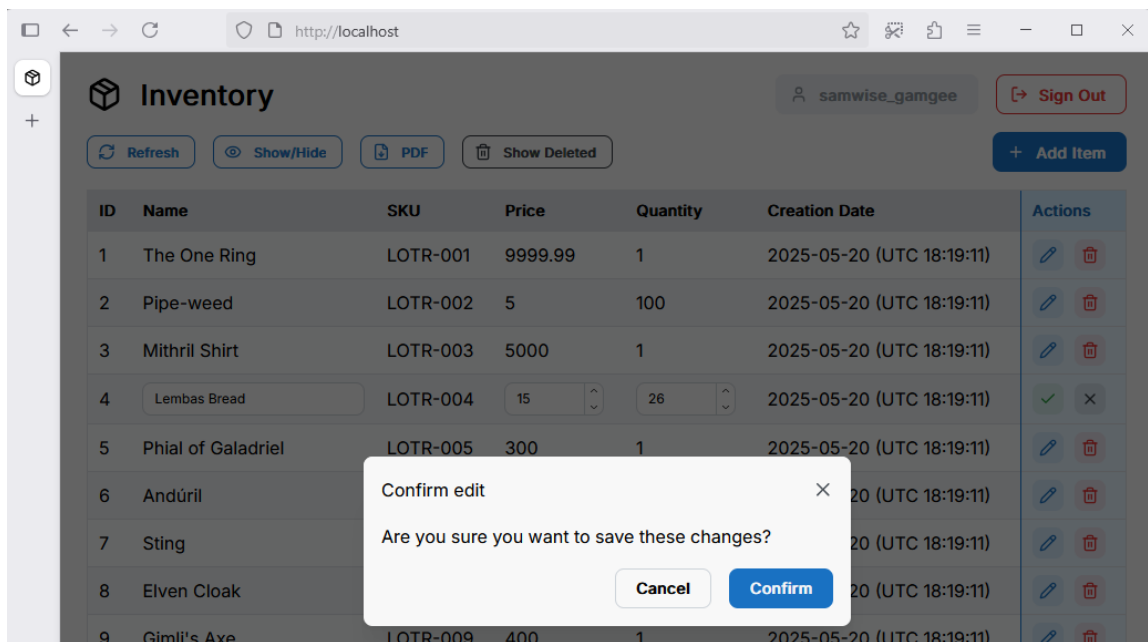


Editar ítem

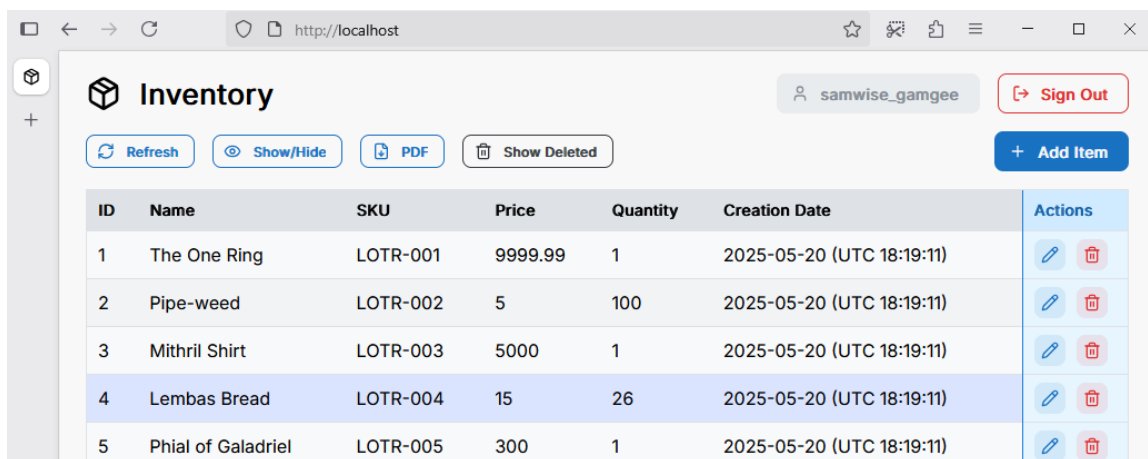
Para editar un ítem existente, se debe hacer clic en el botón de editar correspondiente a la fila del ítem que se desea modificar. Después de hacerlo, la fila cambiará, permitiendo editar los campos que se permiten modificar (todos a excepción de ID, SKU y Creation Date).



Una vez que se han hecho los cambios deseados, se pueden cancelar o enviar haciendo clic en los botones correspondientes, al final de la fila. Si enviamos los cambios, se solicita una confirmación.

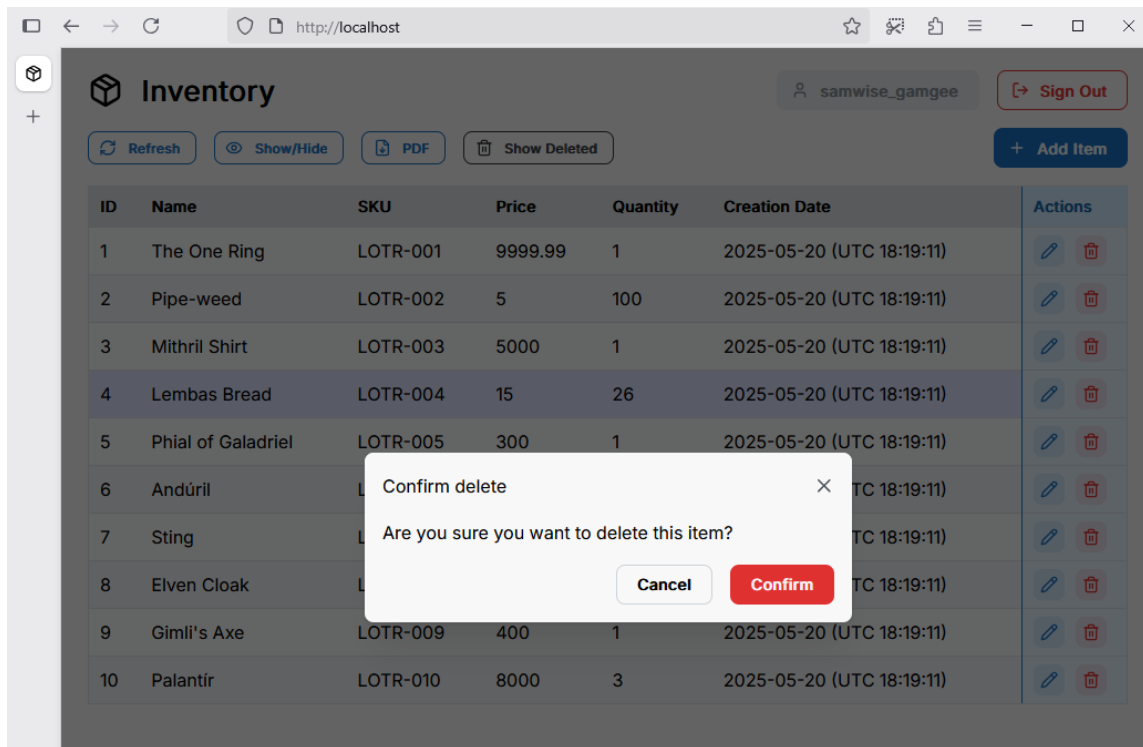


Después de confirmar, podemos verificar que la información cambió y que la fila cambió a color azul.

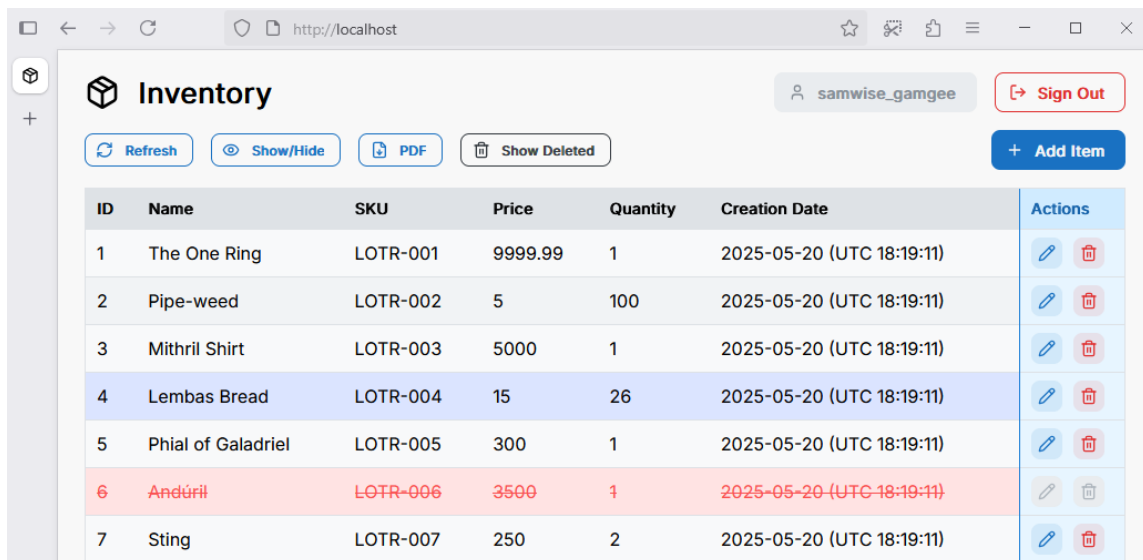


Eliminar ítem

Para eliminar un ítem, se debe hacer clic en el botón de eliminar en la fila correspondiente. Después de hacerlo, se solicitará una confirmación para proceder con la eliminación.

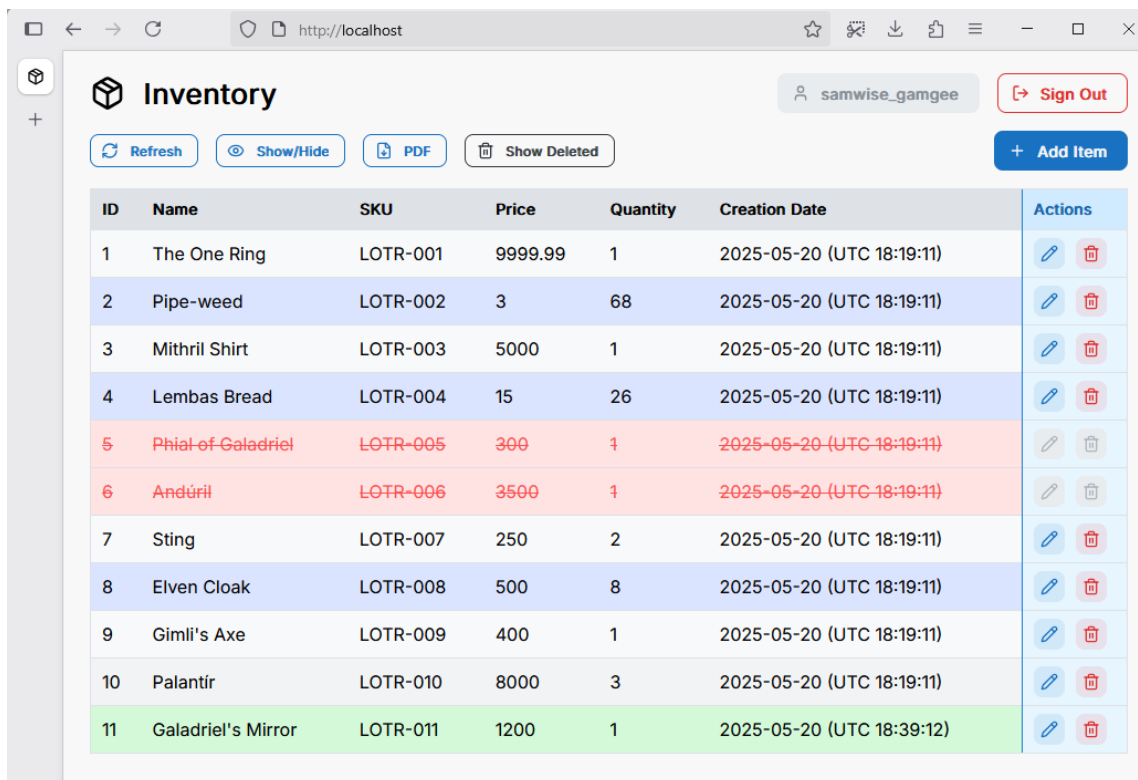


Una vez eliminado, la fila no desaparece, sino que se pone en color rojo con su texto tachado.



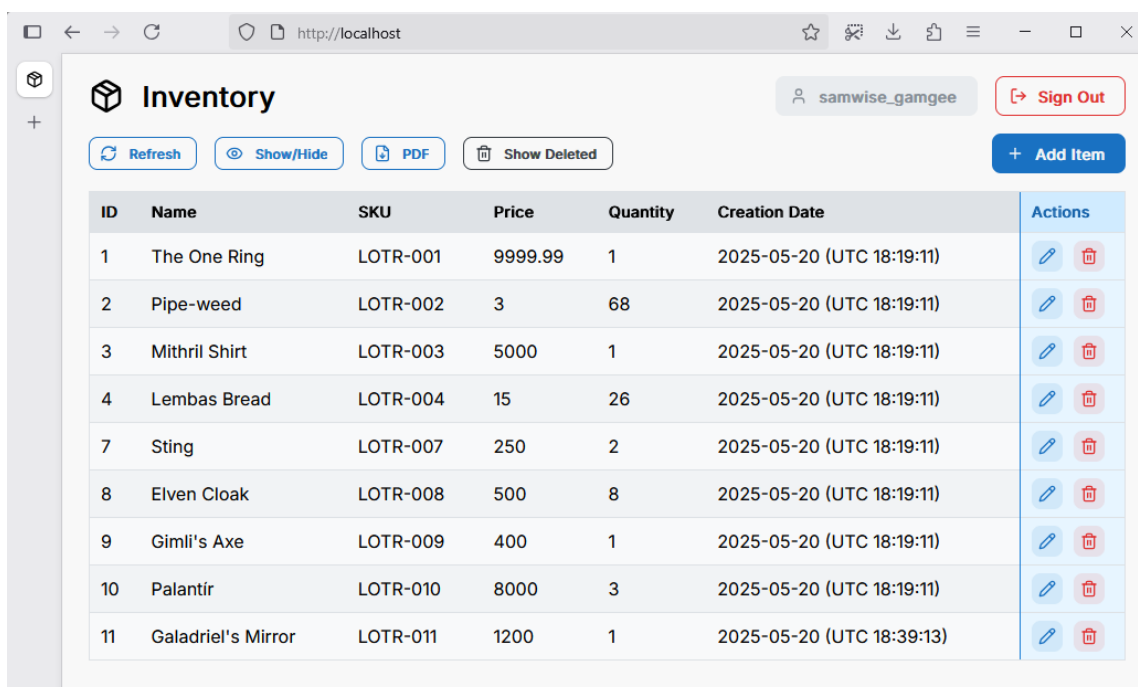
Actualizar la Vista

Hasta ahora, hemos visto que las filas de los objetos agregados, modificados y eliminados cambian de color.



ID	Name	SKU	Price	Quantity	Creation Date	Actions
1	The One Ring	LOTR-001	9999.99	1	2025-05-20 (UTC 18:19:11)	
2	Pipe-weed	LOTR-002	3	68	2025-05-20 (UTC 18:19:11)	
3	Mithril Shirt	LOTR-003	5000	1	2025-05-20 (UTC 18:19:11)	
4	Lembas Bread	LOTR-004	15	26	2025-05-20 (UTC 18:19:11)	
5	Phial of Galadriel	LOTR-005	300	1	2025-05-20 (UTC 18:19:11)	
6	Andúril	LOTR-006	3500	1	2025-05-20 (UTC 18:19:11)	
7	Sting	LOTR-007	250	2	2025-05-20 (UTC 18:19:11)	
8	Elven Cloak	LOTR-008	500	8	2025-05-20 (UTC 18:19:11)	
9	Gimli's Axe	LOTR-009	400	1	2025-05-20 (UTC 18:19:11)	
10	Palantir	LOTR-010	8000	3	2025-05-20 (UTC 18:19:11)	
11	Galadriel's Mirror	LOTR-011	1200	1	2025-05-20 (UTC 18:39:12)	

Estos colores y la vista de los ítems ya eliminados son temporales. Para hacerlos desaparecer, se debe hacer clic en el botón “Refresh”.

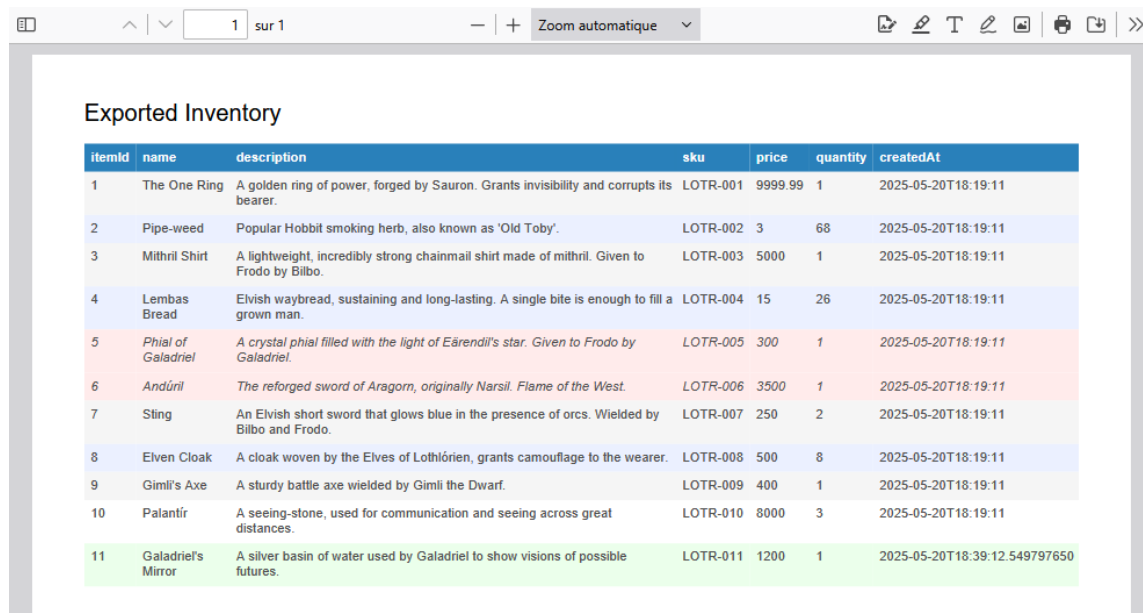


ID	Name	SKU	Price	Quantity	Creation Date	Actions
1	The One Ring	LOTR-001	9999.99	1	2025-05-20 (UTC 18:19:11)	
2	Pipe-weed	LOTR-002	3	68	2025-05-20 (UTC 18:19:11)	
3	Mithril Shirt	LOTR-003	5000	1	2025-05-20 (UTC 18:19:11)	
4	Lembas Bread	LOTR-004	15	26	2025-05-20 (UTC 18:19:11)	
7	Sting	LOTR-007	250	2	2025-05-20 (UTC 18:19:11)	
8	Elven Cloak	LOTR-008	500	8	2025-05-20 (UTC 18:19:11)	
9	Gimli's Axe	LOTR-009	400	1	2025-05-20 (UTC 18:19:11)	
10	Palantir	LOTR-010	8000	3	2025-05-20 (UTC 18:19:11)	
11	Galadriel's Mirror	LOTR-011	1200	1	2025-05-20 (UTC 18:39:13)	

Este botón también actualizará la vista, mostrando los cambios en el inventario que algún otro usuario haya hecho.

Exportar a PDF

Para exportar la vista actual de la tabla a un archivo PDF, se debe hacer clic en el botón “PDF”. Esto generará un archivo PDF con el contenido de la tabla, que se descargará automáticamente. Esta funcionalidad es especialmente útil cuando todavía tenemos los ítems manipulados en colores, pues el PDF se exporta con esos colores, permitiendo esencialmente guardar un registro de los cambios realizados.



itemId	name	description	sku	price	quantity	createdAt
1	The One Ring	A golden ring of power, forged by Sauron. Grants invisibility and corrupts its bearer.	LOTR-001	9999.99	1	2025-05-20T18:19:11
2	Pipe-weed	Popular Hobbit smoking herb, also known as 'Old Toby'.	LOTR-002	3	68	2025-05-20T18:19:11
3	Mithril Shirt	A lightweight, incredibly strong chainmail shirt made of mithril. Given to Frodo by Bilbo.	LOTR-003	5000	1	2025-05-20T18:19:11
4	Lembas Bread	Elvish waybread, sustaining and long-lasting. A single bite is enough to fill a grown man.	LOTR-004	15	26	2025-05-20T18:19:11
5	Phial of Galadriel	A crystal phial filled with the light of Eärendil's star. Given to Frodo by Galadriel.	LOTR-005	300	1	2025-05-20T18:19:11
6	Andúril	The reforged sword of Aragorn, originally Narsil. Flame of the West.	LOTR-006	3500	1	2025-05-20T18:19:11
7	Sting	An Elvish short sword that glows blue in the presence of orcs. Wielded by Bilbo and Frodo.	LOTR-007	250	2	2025-05-20T18:19:11
8	Elven Cloak	A cloak woven by the Elves of Lothlórien, grants camouflage to the wearer.	LOTR-008	500	8	2025-05-20T18:19:11
9	Gimli's Axe	A sturdy battle axe wielded by Gimli the Dwarf.	LOTR-009	400	1	2025-05-20T18:19:11
10	Palantír	A seeing-stone, used for communication and seeing across great distances.	LOTR-010	8000	3	2025-05-20T18:19:11
11	Galadriel's Mirror	A silver basin of water used by Galadriel to show visions of possible futures.	LOTR-011	1200	1	2025-05-20T18:39:12.549797650

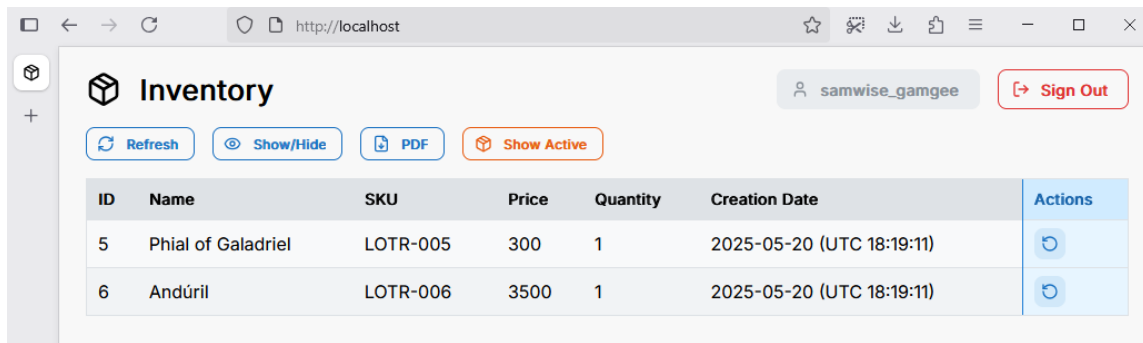
Si las filas no están coloreadas, permanecerán de esa forma.

Exported Inventory

itemId	name	description	sku	price	quantity	createdAt
1	The One Ring	A golden ring of power, forged by Sauron. Grants invisibility and corrupts its bearer.	LOTR-001	9999.99	1	2025-05-20T18:19:11
2	Pipe-weed	Popular Hobbit smoking herb, also known as 'Old Toby'.	LOTR-002	3	68	2025-05-20T18:19:11
3	Mithril Shirt	A lightweight, incredibly strong chainmail shirt made of mithril. Given to Frodo by Bilbo.	LOTR-003	5000	1	2025-05-20T18:19:11
4	Lembas Bread	Elvish waybread, sustaining and long-lasting. A single bite is enough to fill a grown man.	LOTR-004	15	26	2025-05-20T18:19:11
7	Sting	An Elvish short sword that glows blue in the presence of orcs. Wielded by Bilbo and Frodo.	LOTR-007	250	2	2025-05-20T18:19:11
8	Elven Cloak	A cloak woven by the Elves of Lothlórien, grants camouflage to the wearer.	LOTR-008	500	8	2025-05-20T18:19:11
9	Gimli's Axe	A sturdy battle axe wielded by Gimli the Dwarf.	LOTR-009	400	1	2025-05-20T18:19:11
10	Palantír	A seeing-stone, used for communication and seeing across great distances.	LOTR-010	8000	3	2025-05-20T18:19:11
11	Galadriel's Mirror	A silver basin of water used by Galadriel to show visions of possible futures.	LOTR-011	1200	1	2025-05-20T18:39:13

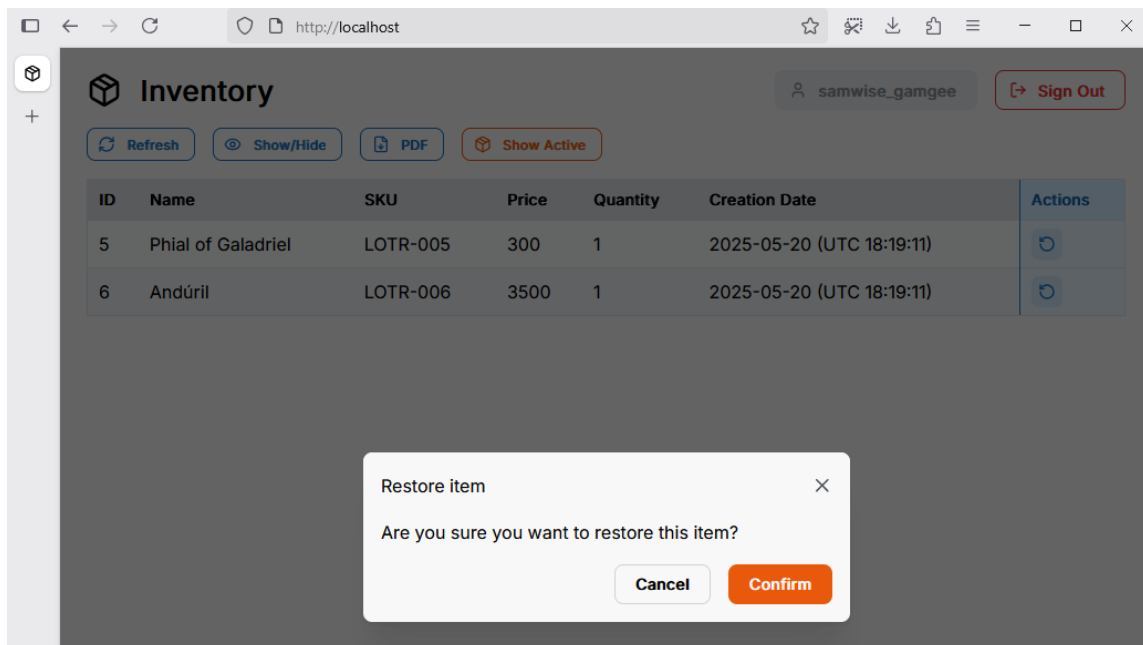
Restaurar ítem

Si queremos ver los ítems que han sido eliminados, basta con darle clic al botón “Show Deleted”.

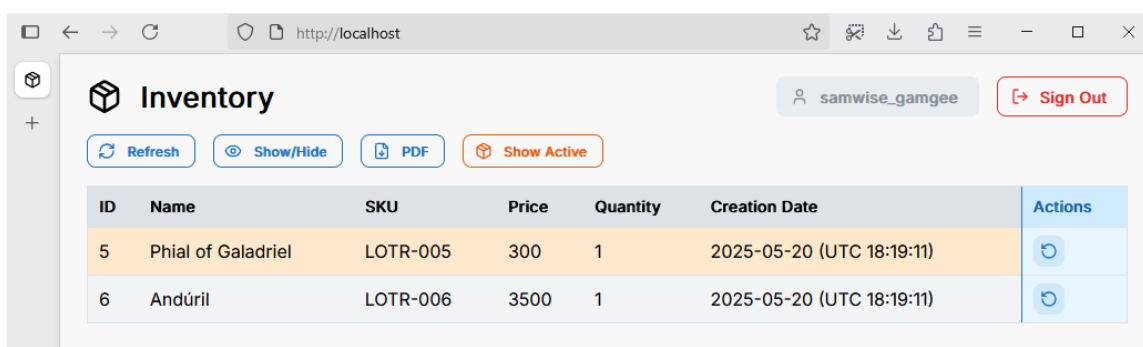


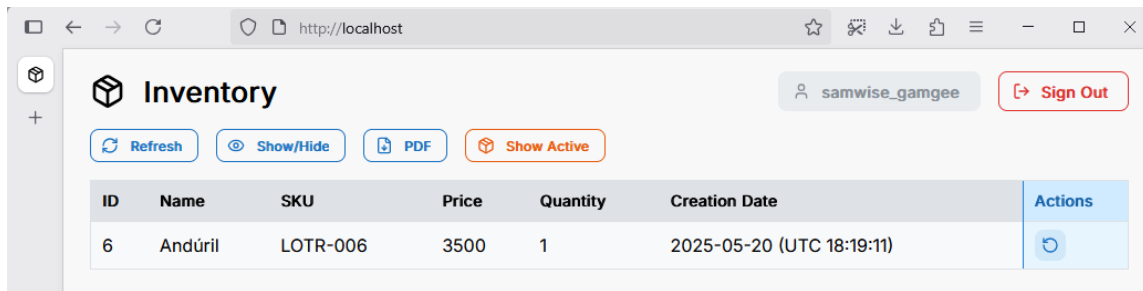
Para regresar a la vista normal de los ítems activos, se da clic al botón “Show Active”.

En esta vista de ítems eliminados, se tiene la opción de restaurarlos (es decir, hacerlos activos de nuevo). Para hacerlo, se da clic al botón de restaurar que aparece al final de cada fila, pidiendo una confirmación.



Los ítems restaurados aparecen en color naranja y, similar a los eliminados en la vista normal, desaparecerán una vez que se actualice la vista.





Conclusión

A lo largo del proyecto, se logró diseñar e implementar una solución web full-stack funcional, segura y escalable, integrando herramientas modernas tanto en el frontend como en el backend.

Entre los principales aprendizajes que se obtuvieron durante el desarrollo de este proyecto destacan la importancia de la planificación arquitectónica (no se puede crear un buen proyecto sin antes planificarlo), la correcta separación de responsabilidades entre los distintos componentes del sistema (¿qué es trabajo del backend y qué del frontend?) y la adopción de buenas prácticas de desarrollo, como el uso de control de versiones (se usó git y GitHub, aprovechando especialmente la funcionalidad de las ramas) y contenedores.

Como áreas de mejora y expansión futura para este proyecto, se identifican:

- Implementar funcionalidades avanzadas, como notificaciones automáticas de bajo inventario o integración con sistemas externos de facturación.
- Mejorar la interfaz de usuario para dispositivos móviles.
- Incorporar pruebas automatizadas.
- Permitir la gestión de múltiples almacenes o sucursales.
- Optimizar el rendimiento y la seguridad ante escenarios de mayor concurrencia.

Estas mejoras no solo enriquecerían la funcionalidad del sistema, sino que también ofrecerían una experiencia de usuario más completa y satisfactoria.

No obstante, no hay que olvidar que el sistema desarrollado cumple con los objetivos planteados: permite a los usuarios gestionar inventarios en línea, realizar operaciones fundamentales sobre los productos y exportar reportes en PDF. Además, la arquitectura modular facilita el mantenimiento y la futura expansión del sistema.